

Parte Practica Capitulo 8

Rafael Puche Q.

Capitulo 8: Reglas de Asociacion

Ejemplo: Identificar compras frecuentes de alimentos

Paso 2: Explorar y preparar la data

Limpiar el workspace

```
rm (list=ls()) setwd(dir = "MEGA_Maestria/Machine_Learning_IVIC/Codes_del_libro/Chapter_08/")
```

Cargar el archivo grocery.csv en una matriz dispersa. Instalacion de paquetes

```
install.packages("arules") install.packages("arulesViz")
```

```
library(arules)
```

```
## Loading required package: Matrix
```

```
##
```

```
## Attaching package: 'arules'
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##      abbreviate, write
```

```
library(arulesViz)
```

```
## Loading required package: grid
```

```
groceries <- read.transactions("groceries.csv", sep = ",")
summary(groceries)
```

```
## transactions as itemMatrix in sparse format with
```

```
## 9835 rows (elements/itemsets/transactions) and
```

```
## 169 columns (items) and a density of 0.02609146
```

```
##
```

```
## most frequent items:
```

```
##      whole milk other vegetables      rolls/buns      soda
```

```
##           2513           1903           1809           1715
```

```
##           yogurt           (Other)
```

```
##           1372           34055
```

```
##
```

```
## element (itemset/transaction) length distribution:
```

```
## sizes
```

```
##      1      2      3      4      5      6      7      8      9     10     11     12     13     14     15
```

```
## 2159 1643 1299 1005 855 645 545 438 350 246 182 117 78 77 55
## 16 17 18 19 20 21 22 23 24 26 27 28 29 32
## 46 29 14 14 9 11 4 6 1 1 1 1 3 1
##
## Min. 1st Qu. Median Mean 3rd Qu. Max.
## 1.000 2.000 3.000 4.409 6.000 32.000
##
## includes extended item information - examples:
## labels
## 1 abrasive cleaner
## 2 artif. sweetener
## 3 baby cosmetics
```

Observar las primeras 5 transacciones

```
inspect(groceries[1:5])
```

```
## items
## 1 {citrus fruit,
## margarine,
## ready soups,
## semi-finished bread}
## 2 {coffee,
## tropical fruit,
## yogurt}
## 3 {whole milk}
## 4 {cream cheese,
## meat spreads,
## pip fruit,
## yogurt}
## 5 {condensed milk,
## long life bakery product,
## other vegetables,
## whole milk}
```

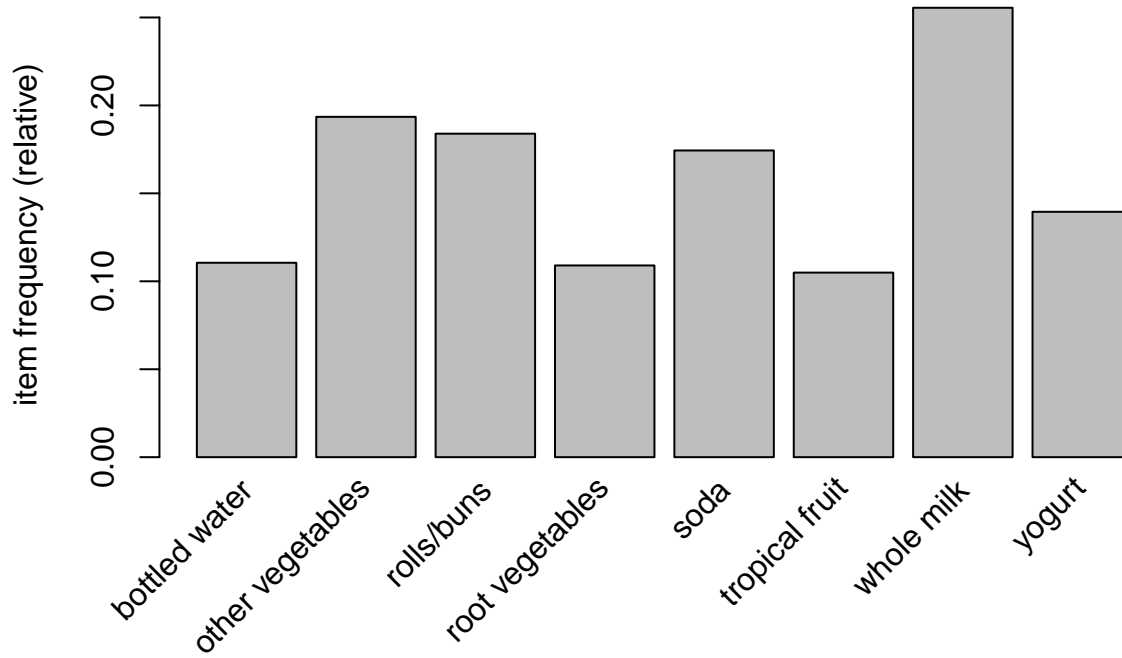
Examinar la frecuencia de los items

```
itemFrequency(groceries[, 1:3])
```

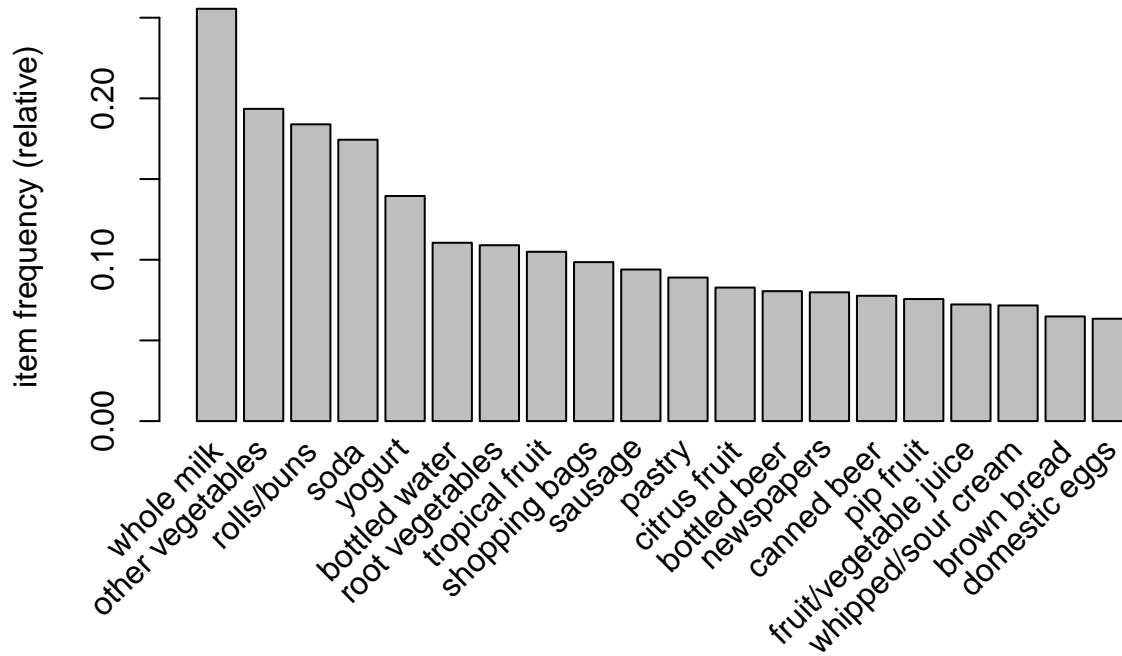
```
## abrasive cleaner artif. sweetener baby cosmetics
## 0.0035587189 0.0032536858 0.0006100661
```

Graficar la frecuencia de los items

```
itemFrequencyPlot(groceries, support = 0.1)
```

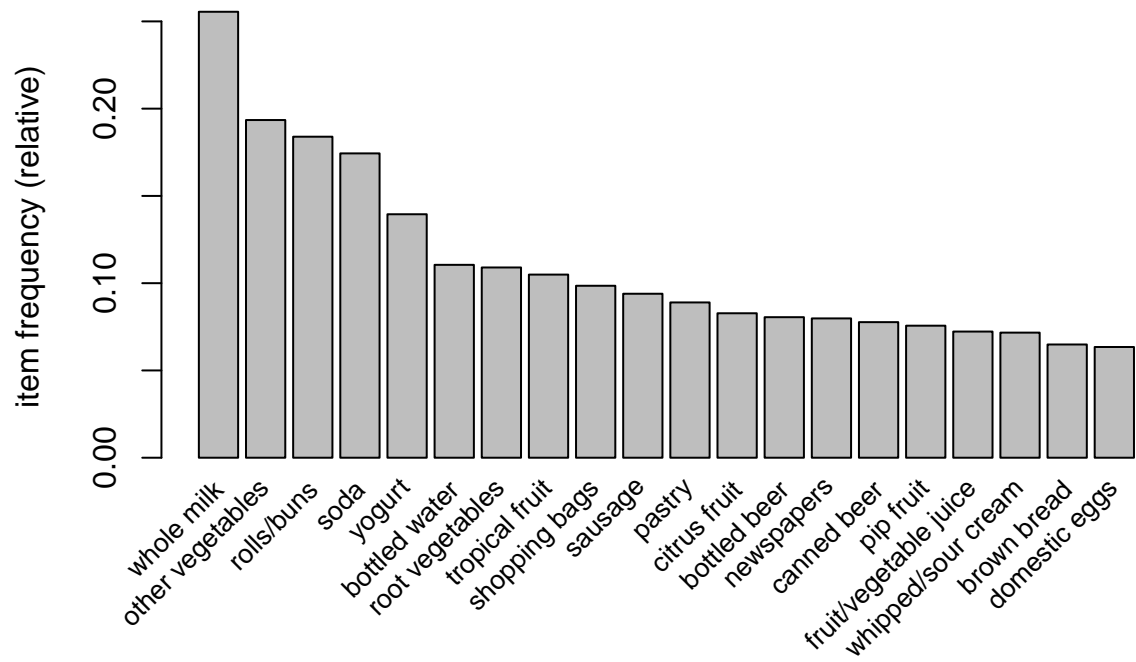


```
itemFrequencyPlot(groceries, topN = 20)
```



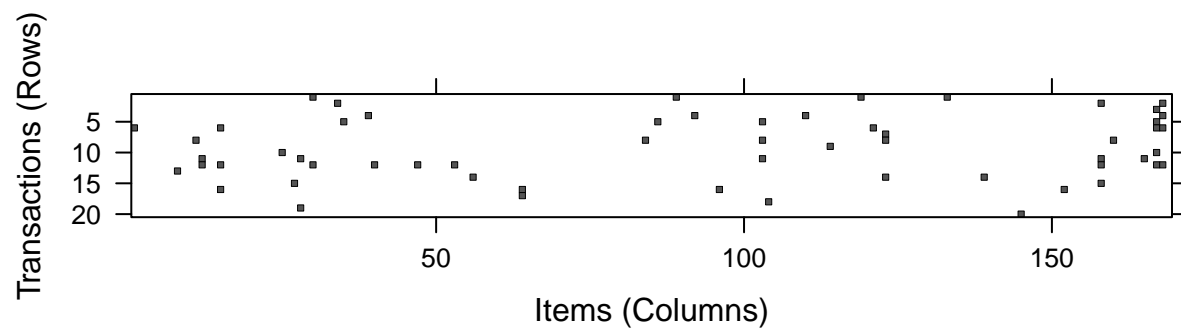
Mejorar la visualizacion de los items con el comando `cex.names`

```
itemFrequencyPlot(groceries, topN = 20, cex.names=0.8)
```



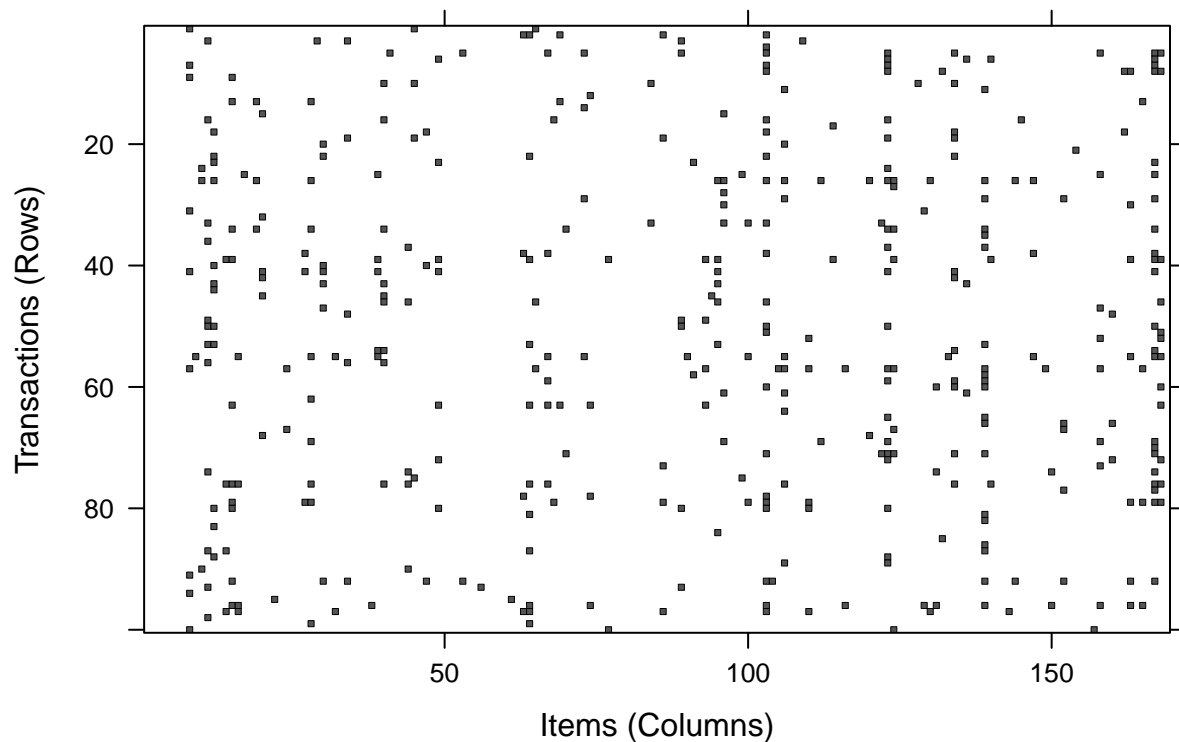
Visualizar la matriz de dispersion de las 5 primeras transacciones

```
image(groceries[1:20])
```



Visualizar una muestra aleatoria de 100 transacciones

```
image(sample(groceries, 100))
```



Paso 3: Entrenar nuestro modelo en los datos

```
library(arules)
```

Condiciones por defecto del algoritmo, generan 0 reglas

```
apriori(groceries)
```

```
## Apriori
##
## Parameter specification:
## confidence minval smax arem aval originalSupport support minlen maxlen
##      0.8      0.1    1 none FALSE          TRUE      0.1      1      10
## target  ext
## rules FALSE
##
## Algorithmic control:
## filter tree heap memopt load sort verbose
##    0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 983
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [8 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
```

```
## checking subsets of size 1 2 done [0.00s].
## writing ... [0 rule(s)] done [0.00s].
## creating S4 object ... done [0.00s].
```

```
## set of 0 rules
```

The appropriate minimum confidence level depends a great deal

on the goals of your analysis. If you start with a conservative

value, you can always reduce it to broaden the search if you aren't

finding actionable intelligence

Ajustar mejor los valores de support y niveles de confianza para aprender mas reglas

minlen = 2 elimina reglas que tienen menos de dos items

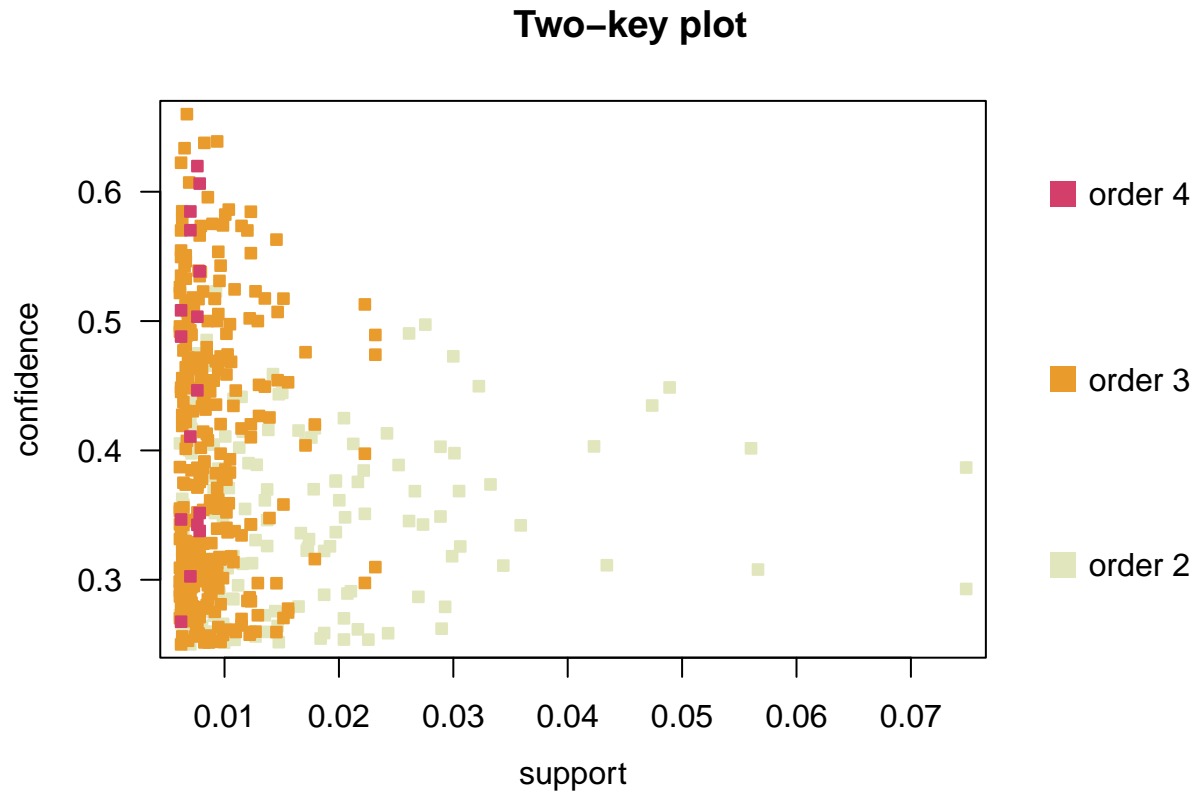
```
groceryrules <- apriori(groceries, parameter = list(support =
  0.006, confidence = 0.25, minlen = 2))
```

```
## Apriori
##
## Parameter specification:
## confidence minval smax arem aval originalSupport support minlen maxlen
##      0.25    0.1    1 none FALSE          TRUE   0.006      2    10
## target  ext
## rules FALSE
##
## Algorithmic control:
## filter tree heap memopt load sort verbose
##    0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 59
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [109 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [463 rule(s)] done [0.00s].
## creating S4 object ... done [0.00s].
```

```
groceryrules
```

```
## set of 463 rules
```

```
plot(groceryrules, shading="order", control=list(main = "Two-key plot"))
```



Prueba de significancia para encontrar reglas de asociacion en las cuales LHS y el RHS dependen uno del otro

```
grocery_sig <- is.significant(groceryrules, groceries, alpha = 0.01, adjust = "bonferroni")
table(grocery_sig)
```

```
## grocery_sig
## FALSE TRUE
##      53  410
```

Ajustando confidence = 0.50 y support = 0.006

```
groceryrules_c50 <- apriori(groceries, parameter = list(support =
  0.006, confidence = 0.50, minlen = 2))
```

```
## Apriori
##
## Parameter specification:
## confidence minval smax arem aval originalSupport support minlen maxlen
##           0.5   0.1   1 none FALSE                TRUE  0.006     2    10
## target  ext
## rules FALSE
##
```

```
## Algorithmic control:
## filter tree heap memopt load sort verbose
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE
##
## Absolute minimum support count: 59
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [109 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [67 rule(s)] done [0.00s].
## creating S4 object ... done [0.00s].

groceryrules_c50

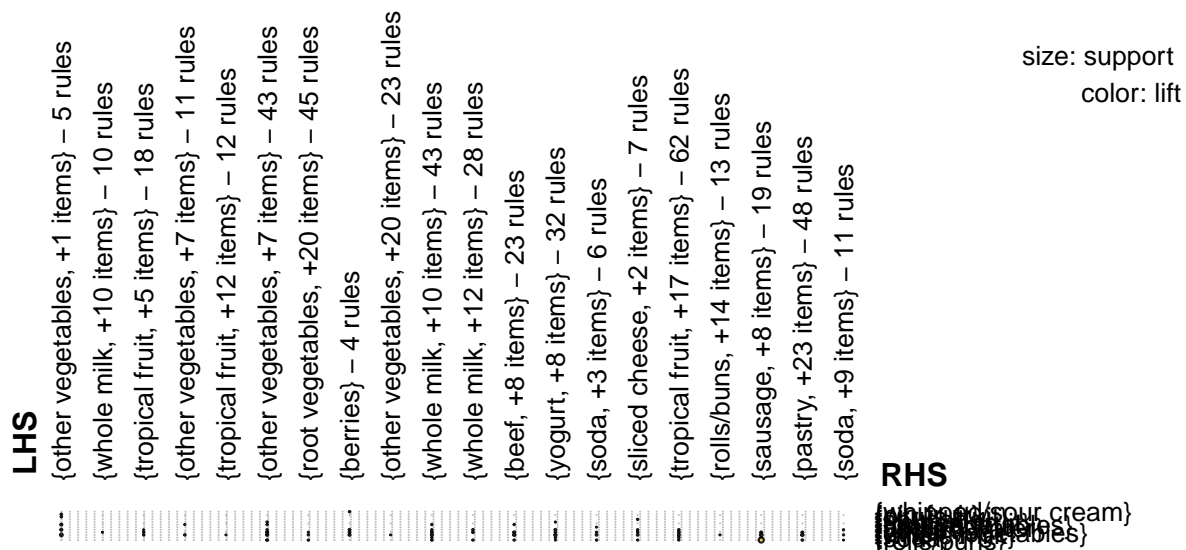
## set of 67 rules

grocery_sig_c50 <- is.significant(groceryrules_c50, groceries, alpha = 0.01, adjust = "bonferroni")
table(grocery_sig_c50)

## grocery_sig_c50
## TRUE
## 67

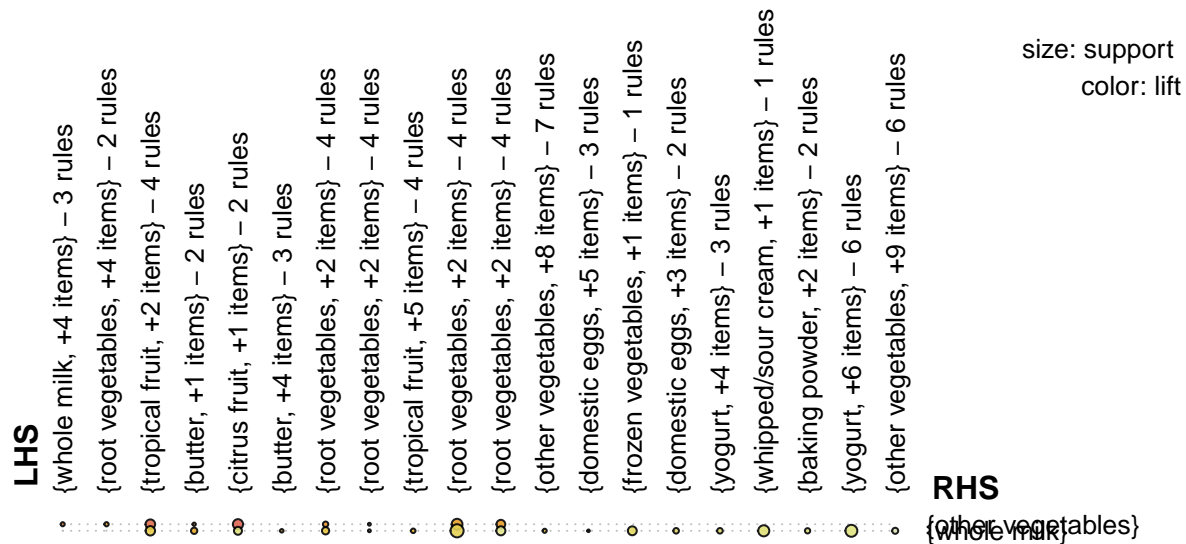
plot(groceryrules, method="grouped", control=list(k=20))
```

Grouped matrix for 463 rules



```
plot(groceryrules_c50, method="grouped", control=list(k=20))
```


Grouped matrix for 67 rules



```
plot(groceryrules, method="matrix", measure=c("lift", "confidence"), control=list(reorder=TRUE))
```

```
## Itemsets in Antecedent (LHS)
## [1] "{other vegetables,tropical fruit}"
## [2] "{other vegetables,tropical fruit,whole milk}"
## [3] "{other vegetables,whole milk}"
## [4] "{other vegetables,rolls/buns,whole milk}"
## [5] "{other vegetables,shopping bags}"
## [6] "{other vegetables,pork}"
## [7] "{domestic eggs,other vegetables}"
## [8] "{frozen vegetables,other vegetables}"
## [9] "{beef,other vegetables}"
## [10] "{herbs}"
## [11] "{onions}"
## [12] "{chicken}"
## [13] "{oil}"
## [14] "{tropical fruit,yogurt}"
## [15] "{beef}"
## [16] "{pork,whole milk}"
## [17] "{domestic eggs,whole milk}"
## [18] "{frozen vegetables,whole milk}"
## [19] "{whipped/sour cream,whole milk}"
## [20] "{butter,whole milk}"
## [21] "{beef,whole milk}"
## [22] "{tropical fruit,whole milk}"
## [23] "{sausage,whole milk}"
## [24] "{other vegetables,sausage}"
## [25] "{other vegetables,soda}"
## [26] "{other vegetables,rolls/buns}"
## [27] "{butter,other vegetables}"
## [28] "{other vegetables,whipped/sour cream}"
## [29] "{fruit/vegetable juice,other vegetables}"
```

```

## [30] "{other vegetables,pip fruit}"
## [31] "{citrus fruit,other vegetables}"
## [32] "{other vegetables,whole milk,yogurt}"
## [33] "{other vegetables,yogurt}"
## [34] "{bottled water,other vegetables}"
## [35] "{other vegetables,root vegetables}"
## [36] "{other vegetables,root vegetables,whole milk}"
## [37] "{citrus fruit,whole milk}"
## [38] "{pip fruit,whole milk}"
## [39] "{whole milk,yogurt}"
## [40] "{whipped/sour cream,yogurt}"
## [41] "{grapes}"
## [42] "{pip fruit}"
## [43] "{citrus fruit,yogurt}"
## [44] "{pip fruit,yogurt}"
## [45] "{root vegetables,yogurt}"
## [46] "{bottled water,yogurt}"
## [47] "{rolls/buns,yogurt}"
## [48] "{butter,yogurt}"
## [49] "{butter,root vegetables}"
## [50] "{domestic eggs,root vegetables}"
## [51] "{pip fruit,root vegetables}"
## [52] "{fruit/vegetable juice,root vegetables}"
## [53] "{citrus fruit,root vegetables}"
## [54] "{frozen vegetables,root vegetables}"
## [55] "{pork,root vegetables}"
## [56] "{beef,root vegetables}"
## [57] "{bottled water,root vegetables}"
## [58] "{root vegetables,soda}"
## [59] "{root vegetables}"
## [60] "{soft cheese}"
## [61] "{sausage,yogurt}"
## [62] "{fruit/vegetable juice,yogurt}"
## [63] "{root vegetables,sausage}"
## [64] "{rolls/buns,whipped/sour cream}"
## [65] "{baking powder}"
## [66] "{curd,yogurt}"
## [67] "{pastry,yogurt}"
## [68] "{flour}"
## [69] "{domestic eggs}"
## [70] "{detergent}"
## [71] "{sugar}"
## [72] "{pickled vegetables}"
## [73] "{pork}"
## [74] "{white bread}"
## [75] "{yogurt}"
## [76] "{pastry,rolls/buns}"
## [77] "{hygiene articles}"
## [78] "{brown bread}"
## [79] "{long life bakery product}"
## [80] "{napkins}"
## [81] "{pastry}"
## [82] "{frozen meals}"
## [83] "{waffles}"

```

```

## [84] "{salty snack}"
## [85] "{whole milk}"
## [86] "{coffee,whole milk}"
## [87] "{sugar,whole milk}"
## [88] "{hamburger meat,whole milk}"
## [89] "{chicken,whole milk}"
## [90] "{fruit/vegetable juice,tropical fruit}"
## [91] "{rolls/buns,root vegetables,whole milk}"
## [92] "{tropical fruit,whole milk,yogurt}"
## [93] "{root vegetables,shopping bags}"
## [94] "{root vegetables,whole milk,yogurt}"
## [95] "{onions,whole milk}"
## [96] "{root vegetables,tropical fruit,whole milk}"
## [97] "{shopping bags,whole milk}"
## [98] "{misc. beverages}"
## [99] "{specialty bar}"
## [100] "{rolls/buns,shopping bags}"
## [101] "{candy}"
## [102] "{bottled water}"
## [103] "{butter,whipped/sour cream}"
## [104] "{curd,tropical fruit}"
## [105] "{butter,tropical fruit}"
## [106] "{other vegetables,tropical fruit,yogurt}"
## [107] "{domestic eggs,tropical fruit}"
## [108] "{other vegetables,root vegetables,yogurt}"
## [109] "{other vegetables,sugar}"
## [110] "{citrus fruit,whipped/sour cream}"
## [111] "{other vegetables,root vegetables,tropical fruit}"
## [112] "{curd,root vegetables}"
## [113] "{frankfurter,yogurt}"
## [114] "{pork,rolls/buns}"
## [115] "{domestic eggs,yogurt}"
## [116] "{margarine,rolls/buns}"
## [117] "{cream cheese,yogurt}"
## [118] "{beef,yogurt}"
## [119] "{sausage,tropical fruit}"
## [120] "{other vegetables,rolls/buns,root vegetables}"
## [121] "{pastry,tropical fruit}"
## [122] "{brown bread,other vegetables}"
## [123] "{beef,rolls/buns}"
## [124] "{napkins,yogurt}"
## [125] "{margarine,yogurt}"
## [126] "{butter,rolls/buns}"
## [127] "{frozen vegetables,yogurt}"
## [128] "{brown bread,yogurt}"
## [129] "{cream cheese,other vegetables}"
## [130] "{coffee,other vegetables}"
## [131] "{napkins,other vegetables}"
## [132] "{bottled beer,other vegetables}"
## [133] "{chicken,other vegetables}"
## [134] "{margarine,other vegetables}"
## [135] "{onions,other vegetables}"
## [136] "{frankfurter,other vegetables}"
## [137] "{hamburger meat,other vegetables}"

```

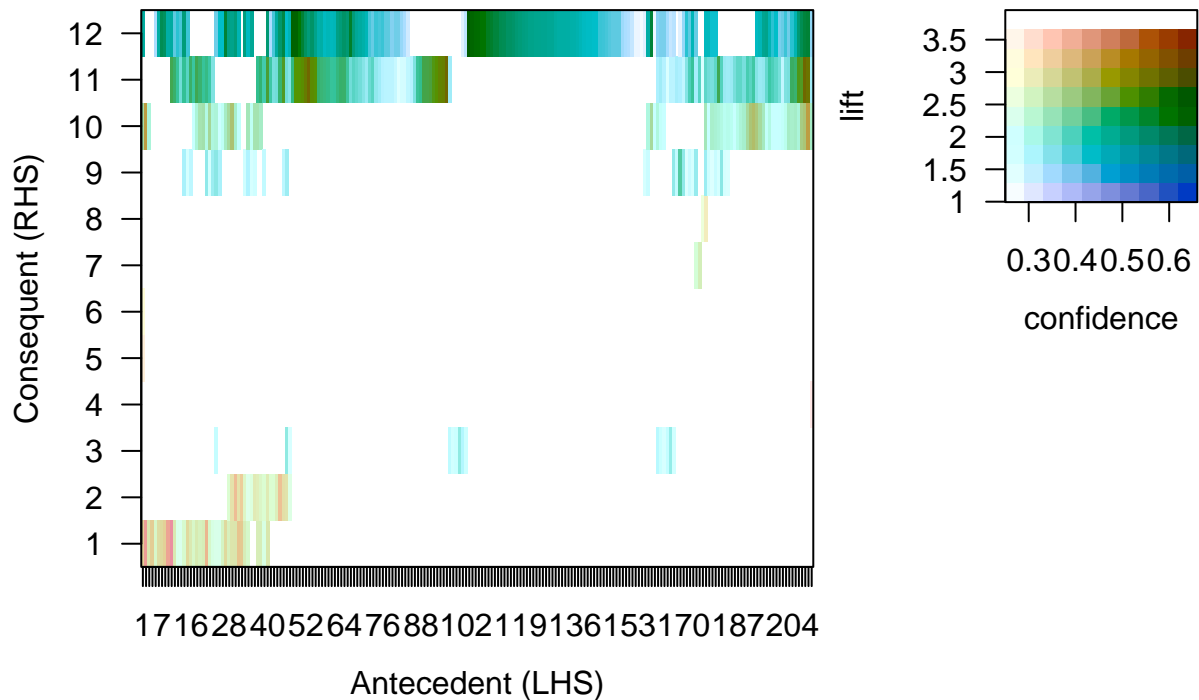
```

## [138] "{pip fruit,rolls/buns}"
## [139] "{newspapers,other vegetables}"
## [140] "{newspapers,yogurt}"
## [141] "{citrus fruit,rolls/buns}"
## [142] "{processed cheese}"
## [143] "{domestic eggs,rolls/buns}"
## [144] "{pasta}"
## [145] "{semi-finished bread}"
## [146] "{potted plants}"
## [147] "{pastry,soda}"
## [148] "{bottled beer,bottled water}"
## [149] "{other vegetables}"
## [150] "{newspapers,rolls/buns}"
## [151] "{newspapers}"
## [152] "{fruit/vegetable juice,soda}"
## [153] "{coffee}"
## [154] "{rolls/buns}"
## [155] "{specialty chocolate}"
## [156] "{beverages}"
## [157] "{bottled beer}"
## [158] "{shopping bags,soda}"
## [159] "{other vegetables,pastry}"
## [160] "{curd,other vegetables}"
## [161] "{bottled water,soda}"
## [162] "{bottled water,rolls/buns}"
## [163] "{dessert}"
## [164] "{fruit/vegetable juice}"
## [165] "{chocolate}"
## [166] "{rolls/buns,sausage}"
## [167] "{sausage}"
## [168] "{newspapers,whole milk}"
## [169] "{sausage,soda}"
## [170] "{frankfurter}"
## [171] "{margarine}"
## [172] "{meat}"
## [173] "{hamburger meat}"
## [174] "{soda,yogurt}"
## [175] "{bottled beer,whole milk}"
## [176] "{rolls/buns,soda}"
## [177] "{sliced cheese}"
## [178] "{ham}"
## [179] "{cream cheese}"
## [180] "{butter milk}"
## [181] "{root vegetables,whole milk}"
## [182] "{margarine,whole milk}"
## [183] "{pastry,whole milk}"
## [184] "{bottled water,whole milk}"
## [185] "{rolls/buns,whole milk}"
## [186] "{soda,whole milk}"
## [187] "{brown bread,whole milk}"
## [188] "{frankfurter,whole milk}"
## [189] "{napkins,whole milk}"
## [190] "{fruit/vegetable juice,whole milk}"
## [191] "{curd,whole milk}"

```

```
## [192] "{cream cheese,whole milk}"
## [193] "{bottled water,tropical fruit}"
## [194] "{rolls/buns,tropical fruit}"
## [195] "{curd}"
## [196] "{butter}"
## [197] "{whipped/sour cream}"
## [198] "{hard cheese}"
## [199] "{frozen vegetables}"
## [200] "{citrus fruit}"
## [201] "{cat food}"
## [202] "{tropical fruit}"
## [203] "{soda,tropical fruit}"
## [204] "{pip fruit,tropical fruit}"
## [205] "{citrus fruit,tropical fruit}"
## [206] "{rolls/buns,root vegetables}"
## [207] "{root vegetables,whipped/sour cream}"
## [208] "{root vegetables,tropical fruit}"
## [209] "{tropical fruit,whipped/sour cream}"
## [210] "{berries}"
## Itemsets in Consequent (RHS)
## [1] "{root vegetables}"      "{tropical fruit}"      "{soda}"
## [4] "{whipped/sour cream}"  "{pip fruit}"          "{citrus fruit}"
## [7] "{bottled water}"      "{sausage}"            "{rolls/buns}"
## [10] "{yogurt}"              "{other vegetables}"    "{whole milk}"
```

Matrix with 463 rules



```
plot(groceryrules_c50, method="matrix", measure=c("lift", "confidence"), control=list(reorder=TRUE))
```

```
## Itemsets in Antecedent (LHS)
```

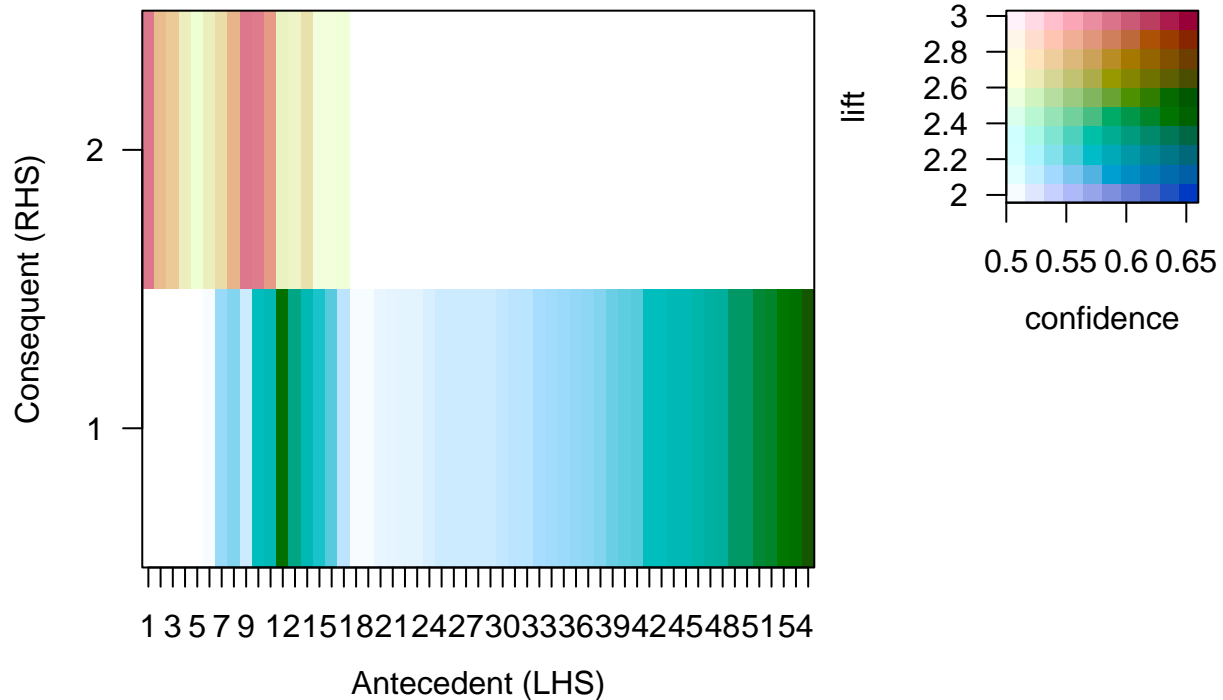
```

## [1] "{root vegetables,tropical fruit,whole milk}"
## [2] "{onions,whole milk}"
## [3] "{root vegetables,whole milk,yogurt}"
## [4] "{root vegetables,shopping bags}"
## [5] "{tropical fruit,whole milk,yogurt}"
## [6] "{pork,root vegetables}"
## [7] "{frozen vegetables,root vegetables}"
## [8] "{fruit/vegetable juice,root vegetables}"
## [9] "{citrus fruit,root vegetables}"
## [10] "{root vegetables,tropical fruit}"
## [11] "{tropical fruit,whipped/sour cream}"
## [12] "{butter,root vegetables}"
## [13] "{domestic eggs,root vegetables}"
## [14] "{pip fruit,root vegetables}"
## [15] "{root vegetables,yogurt}"
## [16] "{root vegetables,whipped/sour cream}"
## [17] "{rolls/buns,root vegetables}"
## [18] "{beef,rolls/buns}"
## [19] "{brown bread,other vegetables}"
## [20] "{fruit/vegetable juice,yogurt}"
## [21] "{other vegetables,whipped/sour cream}"
## [22] "{pastry,tropical fruit}"
## [23] "{other vegetables,rolls/buns,root vegetables}"
## [24] "{other vegetables,yogurt}"
## [25] "{root vegetables,sausage}"
## [26] "{pastry,yogurt}"
## [27] "{tropical fruit,yogurt}"
## [28] "{other vegetables,pip fruit}"
## [29] "{sausage,tropical fruit}"
## [30] "{beef,yogurt}"
## [31] "{baking powder}"
## [32] "{whipped/sour cream,yogurt}"
## [33] "{pip fruit,yogurt}"
## [34] "{cream cheese,yogurt}"
## [35] "{rolls/buns,whipped/sour cream}"
## [36] "{margarine,rolls/buns}"
## [37] "{domestic eggs,yogurt}"
## [38] "{frozen vegetables,other vegetables}"
## [39] "{pork,rolls/buns}"
## [40] "{domestic eggs,other vegetables}"
## [41] "{frankfurter,yogurt}"
## [42] "{curd,root vegetables}"
## [43] "{other vegetables,root vegetables,tropical fruit}"
## [44] "{butter,other vegetables}"
## [45] "{curd,other vegetables}"
## [46] "{citrus fruit,whipped/sour cream}"
## [47] "{curd,yogurt}"
## [48] "{other vegetables,sugar}"
## [49] "{other vegetables,root vegetables,yogurt}"
## [50] "{domestic eggs,tropical fruit}"
## [51] "{other vegetables,tropical fruit,yogurt}"
## [52] "{butter,tropical fruit}"
## [53] "{curd,tropical fruit}"
## [54] "{butter,yogurt}"

```

```
## [55] "{butter,whipped/sour cream}"
## Itemsets in Consequent (RHS)
## [1] "{whole milk}"      "{other vegetables}"
```

Matrix with 67 rules



Paso 4: Evaluar el desempeño del modelo

Resumen de las reglas de asociacion de nuestros datos

```
summary(groceryrules)
```

```
## set of 463 rules
##
## rule length distribution (lhs + rhs):sizes
##   2   3   4
## 150 297  16
##
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  2.000  2.000   3.000   2.711   3.000   4.000
##
## summary of quality measures:
##      support      confidence      lift
##   Min.   :0.006101   Min.   :0.2500   Min.   :0.9932
##   1st Qu.:0.007117   1st Qu.:0.2971   1st Qu.:1.6229
##   Median :0.008744   Median :0.3554   Median :1.9332
##   Mean   :0.011539   Mean   :0.3786   Mean   :2.0351
##   3rd Qu.:0.012303   3rd Qu.:0.4495   3rd Qu.:2.3565
```

```
## Max.      :0.074835   Max.      :0.6600   Max.      :3.9565
##
## mining info:
##      data ntransactions support confidence
## groceries      9835    0.006         0.25
```

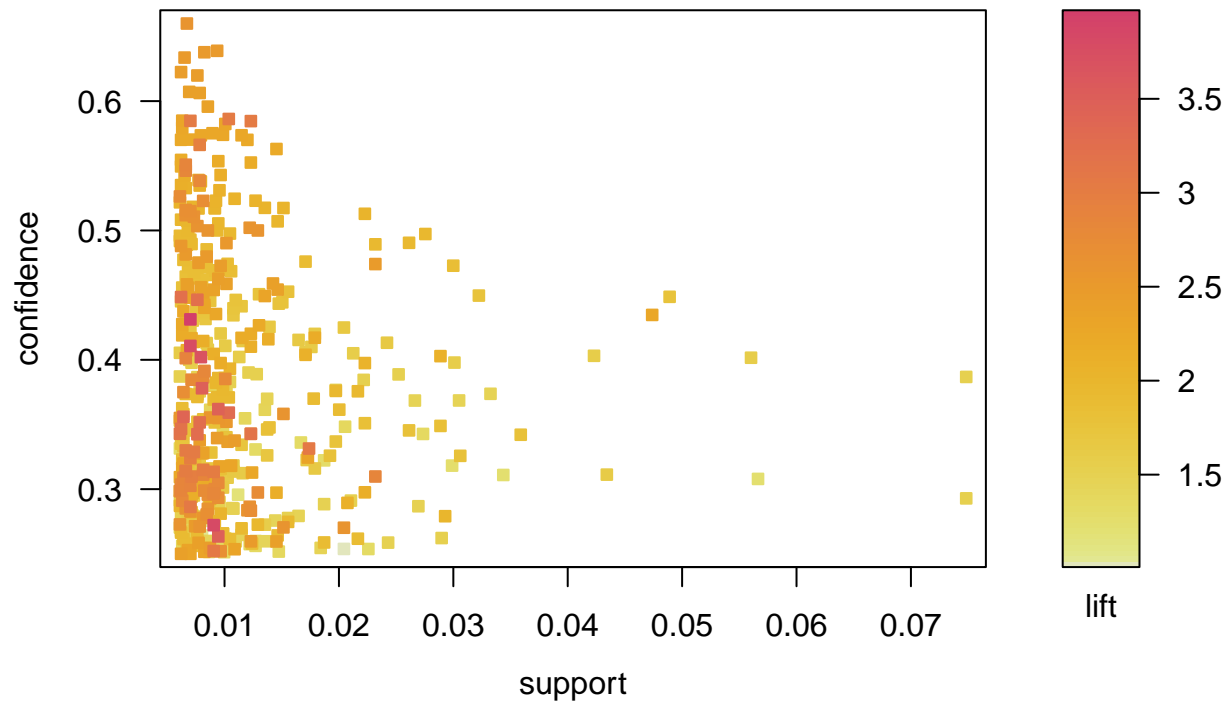
```
summary(groceryrules_c50)
```

```
## set of 67 rules
##
## rule length distribution (lhs + rhs):sizes
##  2  3  4
## 1 59  7
##
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      2.00   3.00   3.00   3.09   3.00   4.00
##
## summary of quality measures:
##      support      confidence      lift
## Min.      :0.006101   Min.      :0.5000   Min.      :1.957
## 1st Qu.:0.006660   1st Qu.:0.5171   1st Qu.:2.050
## Median :0.007829   Median :0.5385   Median :2.232
## Mean    :0.008799   Mean    :0.5485   Mean    :2.316
## 3rd Qu.:0.009609   3rd Qu.:0.5738   3rd Qu.:2.584
## Max.    :0.022267   Max.    :0.6600   Max.    :3.030
##
## mining info:
##      data ntransactions support confidence
## groceries      9835    0.006         0.5
```

Utilizar una libreria para visualizar las reglas de asociacion (arulesViz)

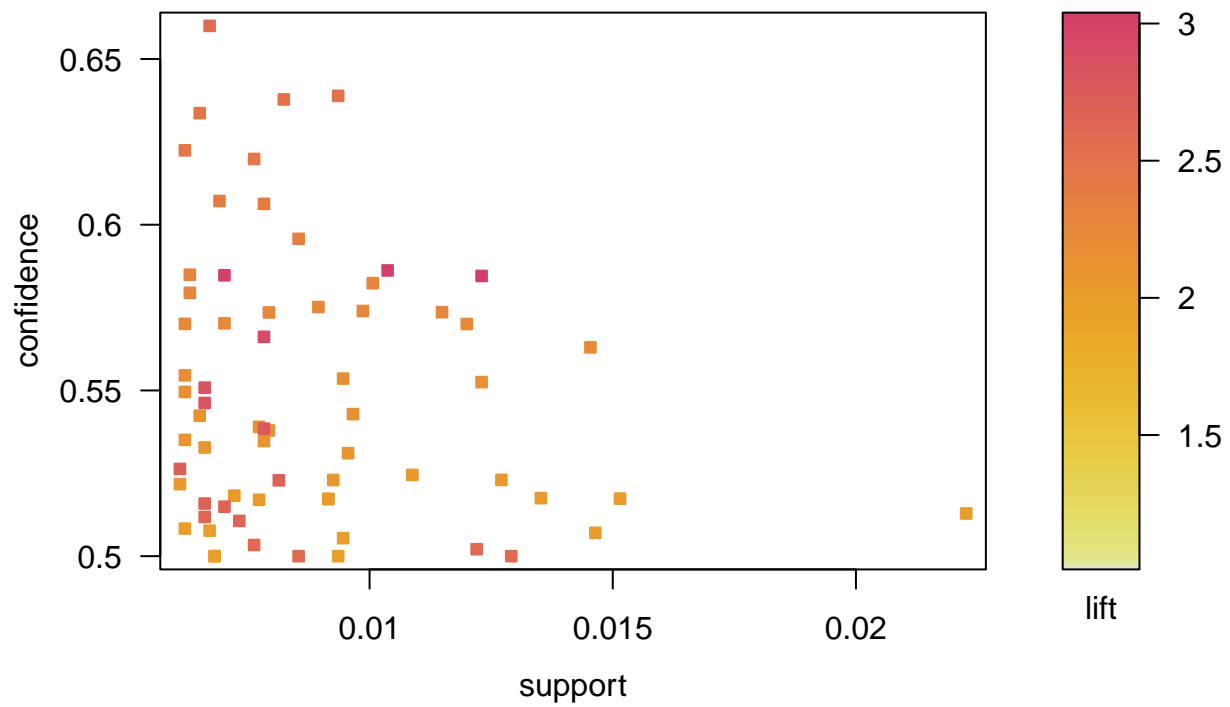
```
plot(groceryrules)
```


Scatter plot for 463 rules



```
plot(groceryrules_c50)
```

Scatter plot for 67 rules



Veamos las tres primeras reglas

```
inspect(groceryrules[1:3])
```

```
##   lhs                rhs                support    confidence lift
## 1 {potted plants} => {whole milk}      0.006914082 0.4000000 1.565460
## 2 {pasta}         => {whole milk}      0.006100661 0.4054054 1.586614
## 3 {herbs}         => {root vegetables} 0.007015760 0.4312500 3.956477
```

```
inspect(groceryrules_c50[1:3])
```

```
##   lhs                rhs                support    confidence
## 1 {baking powder}    => {whole milk}      0.009252669 0.5229885
## 2 {onions,whole milk} => {other vegetables} 0.006609049 0.5462185
## 3 {other vegetables,sugar} => {whole milk} 0.006304016 0.5849057
##   lift
## 1 2.046793
## 2 2.822942
## 3 2.289115
```

Paso 5: Mejorar el desempeño del modelo

Ordenar las “groceryrules” por indice lift mas alto

```
inspect(sort(groceryrules, by = "lift")[1:5])
```

```
##   lhs                rhs                support confidence    lift
## 1 {herbs}            => {root vegetables} 0.007015760 0.4312500 3.956477
## 2 {berries}          => {whipped/sour cream} 0.009049314 0.2721713 3.796886
## 3 {other vegetables,
##   tropical fruit,
##   whole milk}        => {root vegetables} 0.007015760 0.4107143 3.768074
## 4 {beef,
##   other vegetables} => {root vegetables} 0.007930859 0.4020619 3.688692
## 5 {other vegetables,
##   tropical fruit}    => {pip fruit}      0.009456024 0.2634561 3.482649
```

```
inspect(sort(groceryrules_c50, by = "lift")[1:5])
```

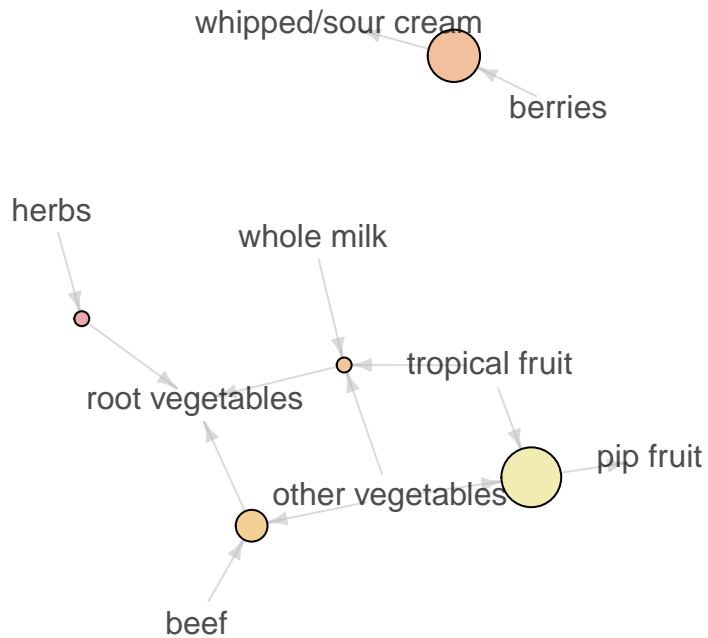
```
##   lhs                rhs                support confidence    lift
## 1 {citrus fruit,
##   root vegetables}    => {other vegetables} 0.010371124 0.5862069 3.029608
## 2 {root vegetables,
##   tropical fruit,
##   whole milk}        => {other vegetables} 0.007015760 0.5847458 3.022057
## 3 {root vegetables,
##   tropical fruit}    => {other vegetables} 0.012302999 0.5845411 3.020999
## 4 {tropical fruit,
##   whipped/sour cream} => {other vegetables} 0.007829181 0.5661765 2.926088
## 5 {fruit/vegetable juice,
##   root vegetables}    => {other vegetables} 0.006609049 0.5508475 2.846865
```

Usar `arulesViz` para visualizar las reglas ordenadas por lift, en esquema vertice y borde

```
sort_grocery <- head(sort(groceryrules, by = "lift")[1:5])
sort_groceryc50 <- head(sort(groceryrules_c50, by = "lift")[1:5])
plot(sort_grocery, method="graph")
```

Graph for 5 rules

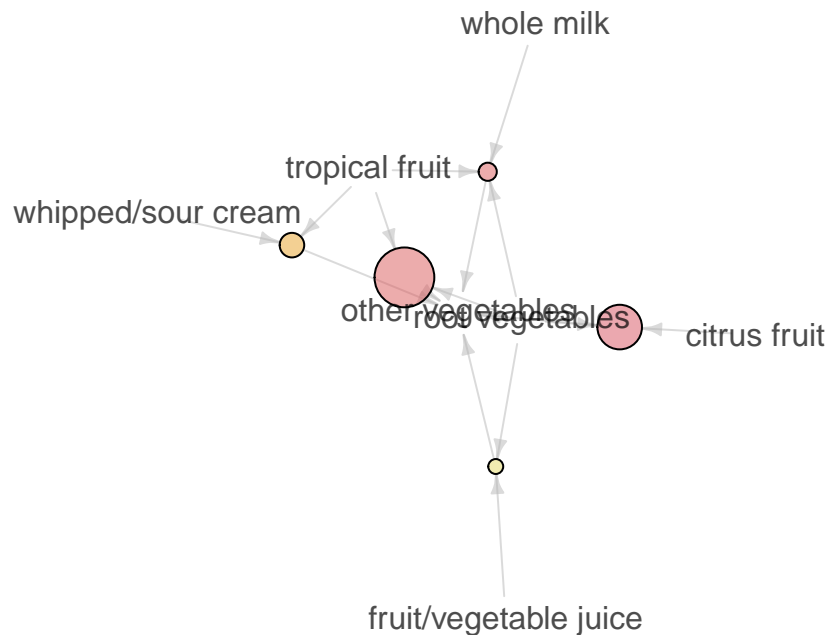
size: support (0.007 – 0.009)
color: lift (3.483 – 3.956)



```
plot(sort_groceryc50, method="graph")
```

Graph for 5 rules

size: support (0.007 – 0.012)
color: lift (2.847 – 3.03)



Encontrar subgrupos de reglas que contienen cualquier item “berries” (fresas, moras, arándanos)

```
berryrules <- subset(groceryrules, items %in% "berries")
inspect(berryrules)
```

##	lhs	rhs	support	confidence	lift
## 57	{berries}	=> {whipped/sour cream}	0.009049314	0.2721713	3.796886
## 58	{berries}	=> {yogurt}	0.010574479	0.3180428	2.279848
## 59	{berries}	=> {other vegetables}	0.010269446	0.3088685	1.596280
## 60	{berries}	=> {whole milk}	0.011794611	0.3547401	1.388328

Escribir las reglas encontradas a un archivo CSV

```
write(groceryrules, file = "groceryrules.csv",
      sep = ",", quote = TRUE, row.names = FALSE)
```

```
#### Convertir el conjunto de reglas en un "data frame"
groceryrules_df <- as(groceryrules, "data.frame")
str(groceryrules_df)
```

```
## 'data.frame': 463 obs. of 4 variables:
## $ rules      : Factor w/ 463 levels "{baking powder} => {other vegetables}",... 340 302 207 206 208 :
## $ support    : num 0.00691 0.0061 0.00702 0.00773 0.00773 ...
## $ confidence : num 0.4 0.405 0.431 0.475 0.475 ...
## $ lift       : num 1.57 1.59 3.96 2.45 1.86 ...
```